Discrete Math Project4

Andreas Griewank

Given N objects of values $0 \leq v_n \in \mathbb{R}$ with weights $0 < w_n \in \mathbb{R}$ and a maximal total weight $0 \leq W \in \mathbb{Z}$ consider the knapsack problem

$$\textbf{maximize} \sum_{n=1}^{N} v_n x_n \quad s.t. \quad \sum_{n=1}^{N} w_n x_n \leq W \quad \text{for} \quad x_n \in \{0,1\} \ .$$

Assume that $w_n = \text{ceil}(.1W\,\text{abs}(\cos(n)))$ and $v_n = \text{ceil}(V\,\text{abs}(\sin(n)))$ or that the weights and values are generated at random from $\{1,\ldots W\}$ and $\{1,\ldots,V\}$ for some $V \geq 1$, respectively.

1. For $N = 300$, $W = 1000$ and $V = 10$ in your favorite programming language:
   a. Use a sorting algorithm to reorder the objects such that the relative value v_n/w_n is monotonically nonincreasing.
   b. Implement the greedy algorithm of taking the first k objects such that the sum of the weights is no greater W but adding the $k+1st$ object would exceed the weight limit.
   c. Derive from the solution of b. the solution of the LOP relaxation, where $x_n \in \{0,1\}$ is replaced by $0 \leq x_n \leq 1$.
2. Formulate the problem as an AMPL model.
   a. Generate the AMPL data file for $N = 300$, $W = 1000$ and $V = 10$ and above.
   b. Solve the problem with Minos or another solver and compare the result to 1.b.
   c. Solve the LOP relaxation with Minos or another solver and compare the result to 1.c.
   d. Round the result of c. by setting all $x_n < 1$ to 0 and compare it to the result of b.
   e. Check whether the solvers benefit from the sorting suggested in 1.a .
   f. Check whether the solvers benefit from an initialization by the greedy solution of 1b.
3. Solve the problem using dynamic optimization.
   a. In your favorite programming language write a program that solves the knapsack problem based on the recurrence

      $\text{maxval}(w,n) = \max(\text{maxval}(w,n-1), v_n + \text{maxval}(w-w_n,n-1))$ for $w \leq W$ and $n \leq N$
   b. Apply the program for $N = 300$, $W = 1000$ and $V = 10$ and compare its results to 2.b.
   c. Modify the program s.t. $\text{maxval}(w,n)$ is not reevaluated for the same argument w,n.
   d. Check whether the new program benefits by the reordering suggested in 1.a .
   e. Apply the program of c. to the standard problem with $N = 300$, $W = 1000$ and $V = 10$ and compare its runtime to that of 2 b. for various solvers.
   f. Run the last program for a large variety of knapsack problems and monitor the ratio of the runtime divided by (NW), is it uniformly bounded?